# AMENDMENTS TO THE CLAIMS

Claims 1-29: Canceled.

30. (currently amended) A method for processing an extensible mark up language (XML) document comprising:

parsing the XML document into a stream of schema elements and data elements;

receiving the stream of schema elements using an API;

converting the stream of schema elements into data type definition (DTD) objects;

validating the stream of data elements using the DTD objects; and

if valid, passing the stream of data elements to an application using the API ~~constructing an in-memory tree representation of the XML document using the data elements~~.

31. (previously presented) The method of claim 30, wherein the converting comprises:

calling a method in a first application program interface (API); and

as a result of calling the first method, calling one or more methods in a second API to construct the DTD objects.

32. (previously presented) The method of claim 30, wherein the converting comprises referencing one or more tables that define the schema elements and associated functions for processing the schema elements.

33. (previously presented) A computer-readable medium having computer-executable instruction, which when executed by a computer, performs the method of claim 30.

34. (currently amended) An architecture for processing an extensible mark up language (XML) document comprising:

a parser to parse the XML document into a stream of elements including a stream of schema elements and a stream of data elements;

a converter to convert the stream of schema elements into data type definition (DTD) objects using an API and to validate the stream of data elements using the DTD objects ~~schema node factory, called by the parser, to handle calls to construct a node in an in-memory tree representation of the XML document for the elements~~; and

a schema node factory to pass valid data elements to an application using the API ~~builder, called by the schema node factory, to construct data type definition (DTD) objects used in validating the data elements~~.

35. (currently amended) The architecture of claim 34, ~~wherein the~~ further comprising a schema builder that utilizes one or more tables to process the elements, the tables containing information defining a schema for the XML data.

36. (previously presented) A computer implemented with the architecture of claim 34.

37. (previously presented) A client-server system, comprising:

a server;

a client connectable to the server to exchange extensible mark up language (XML) documents;

at least one of the client and the server implementing the architecture of claim 34.

38. (previously presented) The architecture of claim 34, further comprising a validation node factory to evaluate whether the data elements comply with constraints set forth in the DTD objects.

39. (currently amended) A system for processing an extensible mark up language (XML) document comprising:

means for parsing the XML document into a stream of schema elements and a stream of data elements;

means for converting the schema elements into data type definition (DTD) objects using an API;

means for validating the data elements using the DTD objects; and

if valid, means for passing valid data elements to an application using the API ~~constructing an in-memory tree representation of the XML document using the data elements~~.